

```
import pandas as pd
from google.colab import files

uploaded = files.upload()
df = pd.read_csv('NVIDIA_STOCK.csv')
df.head()
```



Choose files

No file chosen

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

Saving NVIDIA_STOCK.csv to NVIDIA_STOCK (1).csv

	Price	Adj Close	Close	High	Low
0	Ticker	NVDA	NVDA	NVDA	NVDA
1	Date	NaN	NaN	NaN	NaN
2	2018-01-02	4.929879665374756	4.983749866485596	4.987500190734863	4.862500190734863
3	2018-01-03	5.254334926605225	5.3117499351501465	5.34250020980835	5.09375
4	2018-01-04	5.2820329666137695	5.339749813079834	5.451250076293945	5.317249774932861

```
# Drop the first two header rows
df = df.drop([0, 1]).reset_index(drop=True)

# Rename columns
df.columns = ['Date', 'Adj Close', 'Close', 'High', 'Low', 'Open', 'Volume']

# Convert to appropriate types
numeric_cols = ['Adj Close', 'Close', 'High', 'Low', 'Open', 'Volume']
df[numeric_cols] = df[numeric_cols].apply(pd.to_numeric, errors='coerce')
df['Date'] = pd.to_datetime(df['Date'], errors='coerce')

# Drop missing values
df = df.dropna().reset_index(drop=True)

df.info()
```



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1697 entries, 0 to 1696
Data columns (total 7 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Date        1697 non-null  datetime64[ns]
1   Adj Close   1697 non-null  float64
2   Close       1697 non-null  float64
3   High        1697 non-null  float64
```

```

4    Low      1697 non-null   float64
5    Open     1697 non-null   float64
6    Volume   1697 non-null   int64
dtypes: datetime64[ns](1), float64(5), int64(1)
memory usage: 92.9 KB

```


```
from sklearn.preprocessing import StandardScaler
```

```

scaler = StandardScaler()
df_zscore = df.copy()
df_zscore[numeric_cols] = scaler.fit_transform(df_zscore[numeric_cols])

df_zscore.head()

```



	Date	Adj Close	Close	High	Low	Open	Volume
0	2018-01-02	-0.680102	-0.679439	-0.681400	-0.681068	-0.681750	-0.554661
1	2018-01-03	-0.668996	-0.668209	-0.669489	-0.672972	-0.674635	2.117030
2	2018-01-04	-0.668048	-0.667251	-0.665840	-0.665147	-0.664663	0.533210
3	2018-01-05	-0.666515	-0.665701	-0.666796	-0.666556	-0.666005	0.518186
4	2018-01-08	-0.660928	-0.660052	-0.660010	-0.659991	-0.660694	1.957002

```

df_encoded = df.copy()

# Extract date components
df_encoded['Year'] = df_encoded['Date'].dt.year
df_encoded['Month'] = df_encoded['Date'].dt.month
df_encoded['DayOfWeek'] = df_encoded['Date'].dt.dayofweek

# One-hot encode
df_encoded = pd.get_dummies(df_encoded, columns=['Year', 'Month', 'DayOfWeek'])

df_encoded.head()

```

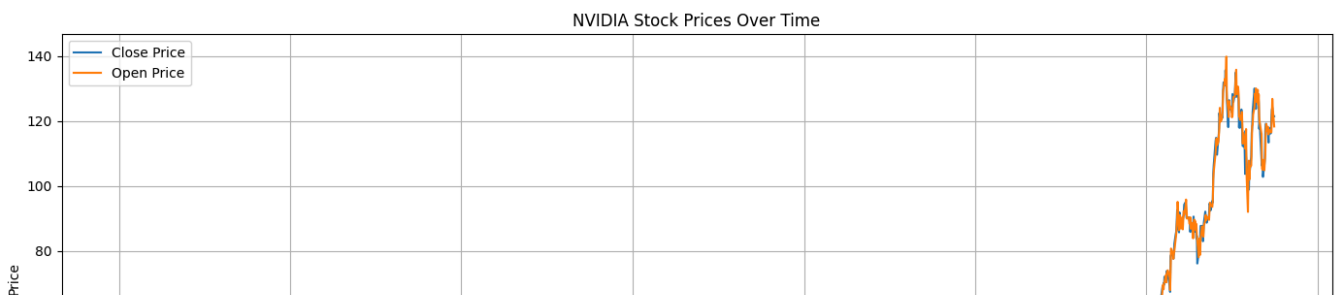


	Date	Adj Close	Close	High	Low	Open	Volume	Year_2018	Year_2019
0	2018-01-02	4.929880	4.98375	4.98750	4.86250	4.89450	355616000	True	False
1	2018-01-03	5.254335	5.31175	5.34250	5.09375	5.10250	914704000	True	False
2	2018-01-04	5.282033	5.33975	5.45125	5.31725	5.39400	583268000	True	False
3	2018-01-05	5.326794	5.38500	5.42275	5.27700	5.35475	580124000	True	False
4	2018-01-08	5.490012	5.55000	5.62500	5.46450	5.51000	881216000	True	False

5 rows × 31 columns

```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(14, 6))
sns.lineplot(data=df, x='Date', y='Close', label='Close Price')
sns.lineplot(data=df, x='Date', y='Open', label='Open Price')
plt.title('NVIDIA Stock Prices Over Time')
plt.xlabel('Date')
plt.ylabel('Price')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```



```
plt.figure(figsize=(14, 4))
sns.lineplot(data=df, x='Date', y='Volume', color='purple')
plt.title('NVIDIA Stock Volume Over Time')
```

```
plt.xlabel('Date')
plt.ylabel('Volume')
plt.grid(True)
plt.tight_layout()
plt.show()
```

